Office de la propriété intellectuelle du Canada

Un organisme d'Industrie Canada

Canadian Intellectual Property Office

An Agency of Industry Canada

CERTIFIED COPY OF PRIORITY DOCUMENT

OIPE MAY 23 2000 PATENT & TRADEMARK OFFICE

Bureau canadien des brevets

Certification

Canadian Patent Office

Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,279,728,** on August 6, 1999, by **SPACEBRIDGE NETWORKS CORPORATION,** assignee of Eko Adi Wibowo and Jun Huang, for "Soft, Prioritized Early Packet Discard (SPEPD) System".

S. Grégoire

Agent certificateur/Certifying Officer

February 17, 2000

Date

BEST AVAILABLE COPY

Canada

(CIPO 68)

OPIC CIPO

ABSTRACT

SPEPD is a congestion control method suitable for satellite on-board switch implementation, but its principle can be applied in various switches, terminals and gateways. The method is especially designed for the access cards of the on-board switch where design simplicity is a critical requirement. SPEPD is intended for Guaranteed Frame Rate (GFR), Available Bit Rate (ABR), Unspecified Bit Rate (UBR) and the forthcoming Weighted UBR (WUBR) service classes.

TITLE OF INVENTION

Soft, Prioritized Early Packet Discard (SPEPD) System

BACKGROUND OF THE INVENTION

Future satellite Asynchronous Transfer Mode (ATM) networks will not be immune to congestion situations because of bursty traffic and a high delay-bandwidth product. As such, it is crucial that an effective congestion control scheme be implemented in such networks.

Congestion durations can vary from cell-level (i.e., on the order of cell transmission time) to call-level (i.e., on the order of call duration). Specific to cell-level congestion, a combination of a queuing discipline and a cell discard scheme is commonly utilized to minimize the intensity and duration of congestion.

A wealth of cell discard schemes are known in the art. The simplest of such scheme is the Discard Tail (DT) scheme, as described in Hashem E., "Analysis of Random Drop for Gateway Congestion Control", Report LCS TR-465, Laboratory for Computer Science, MIT, Cambridge, MA, 1989, where the switch simply discards cells arriving to a full buffer. This scheme has a major weakness in that receivers tend to request retransmission of incomplete packets which contributes to further congestion. Partial Packet Discard (PPD), as described in G. J. Armitage, "Packet Reassembly During Cell Loss", IEEE Network, Vol. 7, No. 9, September 1993 is a scheme that is designed to alleviate the negative effect of re-transmission. In PPD, when a cell is discarded, subsequent cells of the same packet are also discarded, as a result, more buffers are available for whole packets. Early Packet Discard (EPD), as described in Allyn Romanov and Sally Floyd. Dynamics of TCP Traffic over ATM Network. IEEE JSAC, Vol. 13 No. 4, May 1995, pg. 633-641, enhances PPD by discarding whole, instead of partial, packets so that buffer space is used only by whole packets.

In EPD, a threshold on the buffer size is set, above which the switch starts to discard complete packets. Although EPD performs better than PPD, its performance depends on

proper selection of buffer threshold. Buffer threshold should be selected to avoid wasting buffer space (if the threshold is set too low) or the partial discarding of a packet (if the threshold is set too high). The expected number of active connections and the average packet size typically determine this threshold. Unfortunately, both numbers are difficult to estimate.

The use of fixed threshold has another disadvantage with respect to Transmission Control Protocol (TCP). TCP detects congestion only after a packet has been dropped. In EPD, due to its fixed threshold, the switch tends to discard packets successively, resulting in many TCP connections reducing their window size at the same time (i.e., global synchronization). Since the speed of TCP congestion recovery depends heavily on the round-trip delay, the effect of global synchronization to throughput is more significant in satellite networks than in terrestrial networks.

Random Early Discard (RED), is described in Sally Floyd and Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, August 1993, and in B. Braden et. al., "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998. This scheme is designed for networks whose underlying transport protocol is TCP-like (i.e., where a dropped packet is sufficient to indicate congestion to the source). Due to the popularity of the Internet, current and future ATM networks are expected to carry a large proportion of Internet Protocol (IP) traffic. As a result, RED has been considered for implementation in ATM switches, as described in Omar Elloumi and Hossam Afifi, "RED Algorithm in ATM Networks", Technical Report, June 1997. In RED, the switch randomly chooses packets to discard. The number of packets discarded is proportional to the average queue size. RED successfully avoids the global synchronization situation and maintains a low average queue size. RED's major weakness is its computation-intensive algorithm. Furthermore, anticipating that the distribution of traffic is random, it is often unnecessary to perform expensive calculation of drop probability to further randomize the discarding of packets.

All of the above schemes share a common problem of unfairness. Specifically, assuming that both non-TCP-like and TCP-like connections use the same service class and that per-class

queuing is used in the switch, the throughput of non-TCP-like connections is expected to be greater than that of TCP-like connections. EPD with Selective Discard (SD) or Fair Buffer Allocation (FBA), is described in Rohit Goyal, "Traffic Management for TCP/IP over Asynchronous Transfer Mode (ATM) Networks" Ph.D. Dissertation, Department of Computer and Information Science, The Ohio State University, 1999. This approach can successfully avoid the problem of unfairness but does so at the cost of an increase in both memory (and memory bandwidth) requirement and complexity due to the use of per-Virtual-Circuit (VC) accounting. Flow RED, which is described in Dong Lin and Robert Morris, "Dynamics of Random Early Detection", Proceedings of the ACM SIGCOMM '97, September 1997, uses a similar approach to provide equal treatment to non-TCP-like and TCP-like connections.

Further sources of prior art information include

- Jun Huang, "Soft-Bounded Policy for Congestion Control of Communication Network," IEEE PRC on CCSP, Vancouver, Canada, 1991
- Hashem E., "Analysis of Random Drop for Gateway Congestion Control", Report LCS TR-465, Laboratory for Computer Science, MIT, Cambridge, MA, 1989.
- G. J. Armitage, "Packet Reassembly During Cell Loss", IEEE Network, Vol. 7, No. 9, September 1993.
- Sally Floyd and Van Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking. August 1993.
- Allyn Romanov and Sally Floyd, "Dynamics of TCP Traffic over ATM Network", IEEE JSAC, Vol. 13 No. 4, May 1995, pg. 633-641.
- Dong Lin and Robert Morris, "Dynamics of Random Early Detection", Proceedings of the ACM SIGCOMM '97, September 1997.
- Omar Elloumi and Hossam Afifi, "RED Algorithm in ATM Networks" Technical Report, June 1997.
- B. Braden et. Al, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

- Rohit Goyal, "Traffic Management for TCP/IP over Asynchronous Transfer Mode (ATM) Networks", Ph.D. Dissertation, Department of Computer and Information Science, The Ohio State University, 1999.

SUMMARY OF THE INVENTION

This invention is concerned with an intelligent cell discard scheme. Its goals include the mitigation of EPD's inefficiency and the throughput-reducing global synchronization problem of TCP. The novel EPD scheme attempts to combine the simplicity of EPD and the effectiveness of RED for TCP-like connections. In addition, it introduces priority to better match the Quality of Service (QoS) requirements associated with a service class.

SPEPD can reduce the amount of buffer space required on a switch, hence cost of a switch, with its active buffer management while providing better performance than conventional EPD. Furthermore, its RED-like method brings about better TCP throughput. In addition, SPEPD is a general-purpose scheme in that it is effective for mixed-traffic environment due to its use of priority and its non-TCP-specific method.

Main advantages of SPEPD are as follows:

1. Fast-response active queue management.

   Unlike the original RED scheme, SPEPD uses a progressively higher exponential queue length averaging parameter for higher instantaneous queue length. This strategy results in a faster reaction to congestion situation, which is critical for a high speed switch or one with a very limited buffer memory. Furthermore, SPEPD uses a more regular means of updating the average buffer size. This strategy provides multiple advantages to RED's strategy of updating the average buffer size at every packet arrival (i.e., at the arrival of the start- or end-of-packet cell), namely:

   - Faster reaction to congestion situation.
   - More regular updates of average buffer size avoid sudden jump or drop in the average value.

- Worst case situation where updates have to be performed rapidly due to successive arrivals of start- or end-of-packet cells can be avoided.

2. Simple hardware implementation.

   Unlike the random drop strategy of RED, SPEPD uses a strategy of dropping packets regularly. As a result, the complexity in calculating the random drop probability as well as in determining whether or not to perform packet discard is greatly reduced. The risk of service unfairness from dropping packets regularly is not an issue in broadband networks since traffic arrival will be random in nature due to the high level of multiplexing.

3. Prioritized packet-level discard.

   SPEPD introduces priority to provide differentiated service among service classes sharing a common buffer pool or among service groups sharing a common service class buffer space. This strategy also provides a base for efficient buffer management for future applications and services, such as Virtual Private Networks (VPN).

4. Flexibility.

   In SPEPD, buffer averaging parameters and packet count thresholds are implemented using lookup table(s). These parameters can then be readily fine-tuned to suit observed traffic characteristics and available buffer space.

SPEPD can also be enhanced with per-Virtual Channel (VC) accounting to provide further fair treatment to both TCP-like and non-TCP-like connections.

SPEPD also improves upon RED's packet counting by halting the count of packets while the average buffer size is below the congestion threshold. This is in contrast to the RED's strategy of resetting the packet counter to zero whenever the average buffer size value is below the minimum threshold. By preventing the discard scheme from an overly slow response to congestion halting the packet counting improves upon the count reset scheme used by RED. The halting of the packet count in SPEPD also prevents an overly rapid response to congestion caused by a continuing count of packets.

A weighting factor, Alpha, is also used by both RED and SPEPD to calculate the average buffer size. Optimal buffer size varies with several factors, including congestion, which

makes a fixed Alpha undesireable. Unlike RED, SPEPD uses different Alpha values for different congestion situation.

The strategy for triggering the update of average buffer size is another strategy differentiator of SPEPD. With RED, the average buffer size is updated whenever a new packet arrives, which is equivalent to updating the average buffer size whenever a start-of-packet cell arrives. Therefore, when start-of-packet cells arrive successively, the average buffer size value will be updated a numerous times in a very short interval, bringing about rapid fluctuations in the value of the average buffer size, which diminishes the accuracy of the average value. The SPEPD scheme avoids this problem by conducting more regular updates of the average value.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1. Packet Count Threshold Function F(n)

Figure 2. Packet Count Threshold Function F(n, m) for Low Priority Class Traffic

Figure 3. Flowchart of Single-Priority SPEPD Scheme

Figure 4. Buffer Size Averaging Method

Figure 5. A Block Diagram of SPEPD for Hardware Implementation

Figure 6. Network Setup for SPEPD Simulations

Figure 7. Parameter Setup for SPEPD Simulation

Figure 8. Cell Loss Ratio for TCP Traffic at the ATM Switch (EPD)

Figure 9. Cell Loss Ratio for TCP Traffic at the ATM Switch (SPEPD)

Figure 10. TCP Delay at Server (EPD)

Figure 11. TCP Delay at Server (SPEPD)

DETAILED DESCRIPTION OF THE INVENTION

SPEPD SCHEME

A prior art method of soft-bounded congestion control, as described below, is modified in accordance with this invention to create a simple dynamic threshold EPD. For connectionless networks such as computer networks, conventional congestion control schemes are based on selectively discarding packets or equivalently adjusting buffer size according to the level of congestion. Under normal circumstances no packets are discarded, but under heavy traffic conditions packet loss is common, and the levels of loss depend upon the weight of the traffic. Similarly for connection oriented networks congestion control schemes are partially based on denying new calls as a result of to the level of congestion in the system. Congestion is not a static resource shortage problem, it is dynamic in nature. Packet loss due to buffer shortage and packet delay due to buffer sufficiency are only symptoms, not causes of congestion. The causes are stochastic properties of the traffic. If the system traffic is deterministic no congestion problems will occur. Therefore, the objective of this novel packet discard scheme is not only to create the resource or to reduce the commands but also to create the dynamic resources and to reduce the fluctuation of the demand.

Let K^* denotes the state of the network without dynamic control of station capacities, $K^* = (k_1^*, k_2^*, \ldots, k_L^*)$ where k^* is the number of packets in the i[th] station. After the control of buffer size is enabled, it is clear that all states K^* cannot be feasible. The idea is to normalize the infeasible states. The packet distribution in each state is adapted to the capacity limit of each station in the blocking network by $f(k^*) = (k)$ where k is the normalised state for the blocking network. The function f transforms the non-feasible state (k^*) to the feasible state (k).

With this soft-bounded congestion control in place, the following modifications are done to create the dynamic threshold EPD in accordance with this invention. The total buffer size can be divided into N regions. When the average buffer size is above a threshold that indicates congestion (henceforth referred to as congestion threshold for conciseness) and is in the region n, $1 \le n \le N$, one of every F(n) packets will be discarded. An exemplary embodiment of

the function F(n) is given in Figure 1. This strategy of discarding packets regularly provides for simple hardware implementation. The risk of service unfairness, that is, the discard of successive packets from a connection, is negligible in the environment where a soft prioritized early packet discard (SPEPD) system in accordance with this invention will find use, i.e., high-speed core network elements. This is because the average buffer occupancy level, the size of packets and the number of active connections fluctuate very quickly. Furthermore, the high level of multiplexing in such network elements has likely randomized the arrival of packets.

As shown in Figure 1, F(n) has higher values for lower average buffer sizes. This is aimed at addressing the problem of global synchronization of TCP connections. By allowing adequate spacing between two packet discards, it is likely that only one TCP connection will be forced to reduce its window size at a time. As a result, the overall TCP throughput will not be significantly affected. On the other hand, the function F(n) has lower values for higher average buffer sizes. This is aimed at making available a certain amount of buffer space to store bursty packets. As a result, partial packet discards, which can quickly trigger the global synchronization problem, can be avoided.

In order to perform packet discard regularly, SPEPD counts the number of packets that have arrived since the last packet discard event. Then, when the number of new packet arrivals is equal to the current value of the function F(n), the newly arriving packet is discarded. The counting of packets is halted whenever the average buffer size value is below the congestion threshold.

The averaging of buffer size is achieved using the exponential averaging scheme much like that of RED. The average buffer size is calculated using the function shown below:

New Average = Old Average * (1 – Alpha) + Current Buffer Size * Alpha

The averaging variable "Alpha" is a weighting factor that expresses the importance of the current, instantaneous buffer size with respect to the average buffer size. A high Alpha value

(close to 1) means the current, instantaneous value is weighted more heavily towards the calculation of the average value, thereby reducing the importance of past instantaneous values. A low Alpha value (close to 0) increases the importance of past instantaneous values at the expense of current ones. Therefore, the degree of fluctuation in the average value is significantly affected by the choice of Alpha.

SPEPD uses a progressively higher value of Alpha as congestion worsens. This strategy is one of the key strategies to this approach and is intended to address the shortcomings of the RED scheme with respect to the speed of response to congestion situation. The SPEPD scheme uses a low value of Alpha when the instantaneous buffer size is low. This is intended to play down the instantaneous buffer size value so that the scheme can react faster to future, more severely congested situations. Conversely, the SPEPD scheme uses a high value of Alpha when the instantaneous buffer size is high. This is intended to increase the importance of the instantaneous buffer size value in order for the scheme to respond quickly to current congestion situation. Furthermore, the increase in speed of response also translates to a reduction in the amount of required buffer space for the scheme to work effectively, which makes the SPEPD scheme suitable for an environment where hardware resources are very limited, such as for a satellite on-board switch.

SPEPD conducts updates of the average value at regular intervals to avoid rapid fluctuations in buffer size. There are two alternatives for achieving this, one is to perform an update whenever a certain number of cells (or packets in packet-based switches) have arrived since the last update. The second alternative is to perform an update whenever a certain amount of time has elapsed since the last update. The first alternative is more suitable for a software implementation of the scheme, while the second alternative is more appropriate for a hardware implementation of the scheme. A flowchart of a scheme to implement SPEPD using the first alternative of triggering the update is given in Figure 3 and 4.

Finally, the SPEPD scheme is optionally augmented with a priority scheme allowing differentiated service among service classes sharing a common buffer pool, or among service groups sharing a common service class buffer space. This priority scheme provides a base for

efficient buffer management for future applications and services, such as Virtual Private Networks (VPN). The following exemplary embodiment describes the method to support the sharing of buffer space between two priority classes of traffic using the SPEPD scheme. The total buffer size is divided into M regions for high priority traffic, and into N regions for low priority traffic. In this case, when the average buffer size of high priority traffic is above the congestion threshold for high priority traffic and is in the region m, $1 \leq m \leq M$, one of every F(m) high priority packets will be discarded. Furthermore, when the average buffer size of low priority traffic is also above the congestion threshold for low priority traffic and is in the region n, $1 \leq n \leq N$, one of every F(n,m) low priority packets will be discarded. Here, the function F(m) is similar to that given in Figure 1, while the function F(n, m) looks like that given in Figure 2.

2. HARDWARE EMBODIMENT

In order to illustrate a hardware embodiment of SPEPD, the single-priority flavour of SPEPD with arrival-based triggering of buffer averaging update is considered. Given a total buffer size divided into N = 16 equal regions, the function F(n) (whose values are denoted as *PkCntThr* in Figure 3) is implemented as a lookup table, with the index being the four Most Significant Bits (MSB) of the average buffer size (i.e., the lookup table stores $2^4 = 16$ elements). Next, associated with each region is one buffer averaging parameter which is denoted as Alpha. These parameters are also stored in a lookup table, with the index being the 4 MSB of the instantaneous buffer size. For ease of implementation, Alpha will be in the form 2^{-k}, where k is an integer (Alpha is generally a small number, such as one of those given in Figure 7; appropriate values of Alpha are found empirically or through software simulations). Finally, PPD is initiated when a cell is discarded due to buffer starvation.

Two-state variables are maintained for each eligible connection and are incorporated into a connection table, which stores all relevant information about each connection, including state variables associated with SPEPD. The first state variable is used to store the state of discard, namely IDLE, EPD and PPD, while the second state variable is used to indicate whether the last cell received from the connection is an end-of-packet cell. In addition, four global

variables are maintained for each class of traffic to which the SPEPD scheme applies. The first global variable is used to count the number of cells buffered (denoted as *BufCnt* in Figure 3 and 4). The second variable is used to store the average number of cells buffered (denoted as *AvBufCnt* in Figure 3 and 4). The third variable is used to count the number of packets seen since the last packet discard (denoted as *PkCnt* in the Figure 3). The fourth variable is used to count the number of cell arrivals since the last buffer size averaging event (denoted as *ClCnt* in Figure 3 and 4). Finally, two fixed parameters are maintained for each class of traffic. The first fixed parameter is the congestion threshold (denoted as *CongThr* in Figure 3). The counting of packets is disabled when the average buffer size is below this threshold. The second fixed parameter is the cell counter threshold (denoted as *ClCntThr* in Figure 4). The new average buffer size value is calculated when the number of cell arrivals is equal to this threshold. The flowchart of a single-priority SPEPD scheme is shown in Figure 3. The calculation of average buffer size is depicted by the flowchart shown in Figure 4.

Finally, a block diagram depicting the hardware implementation of SPEPD is shown in Figure 5. The Write process handles each newly arriving cell. The processing involved includes the appending of internal cell headers and obtaining of buffer address to store the cell. The Write process needs to wait for the SPEPD process to grant the buffering of the new cell and supplies the connection identifier of the new cell to the SPEPD process. The SPEPD process, in turn, utilizes this information along with other information as depicted in Figure 3 and 4 in order to decide whether to store or discard the cell. The Buffer Manager process is in charge of the overall accounting of buffer space. This process supplies the buffer counter values to the SPEPD process. The lookup table stores information regarding packet count thresholds and buffer averaging parameters, which are obtained by the SPEPD by supplying the four MSB's of the average buffer size and four MSB's of the instantaneous buffer size, respectively. Finally, the Read process handles the processing of each outgoing cell.

3. PERFORMANCE

In order to illustrate the advantage of SPEPD over the conventional EPD schemes, performance evaluation through software simulations have been conducted. Simulations are conducted for a network as shown in Figure 6.

In the simulation, each of the 15 workstations is generating File Transfer Protocol (FTP) traffic destined to the server. Each workstation is set to generate 270 TCP sessions per hour, distributed according to the Poisson distribution. Each TCP session is an upload FTP session with an average upload size of 128 KB, again distributed according to the Poisson distribution. The TCP model used is based on the industry standards known as Request For Comment (RFC) 793 and RFC 1122. The propagation delay in each T1 link connecting a workstation or the server with the ATM switch is set to be 125 ms, which is a typical ground-to-satellite delay for Geosynchronous (GEO) satellites. Finally, the average load on the link from the ATM switch to the server is approximately 76 %.

The setup of SPEPD parameters used in the simulation is shown in Figure 7.

Figure 8 and Figure 9 show the cell loss ratio for TCP traffic (transported using UBR service) measured at the ATM switch for the case where EPD and SPEPD, respectively, are used.

Figure 10 and Figure 11 show the average TCP delay measured at the server for the case where EPD and SPEPD, respectively, are used.

These figures confirm the superior performance of SPEPD over the conventional EPD technique. By performing packet discard earlier than EPD and by spacing packet discard over multiple packets, SPEPD manages to alleviate the problem of global TCP synchronization, resulting in shorter congestion period, lower cell loss ratio (CLR is reduced almost by a factor of 2) and lower TCP delay (maximum TCP delay observed during simulation is close to 80 seconds for the case of EPD and close to 50 seconds for the case of SPEPD).

4. APPLICATIONS

This novel packet discard scheme will find application in numerous telecommunication fields such as Traffic Management, ATM switches (both terrestrial and satellite based), and ATM Terminals and Gateways.

CA 02279728 1999-08-06

GLOSSARY OF ACRONYMS

ATM	Asynchronous Transfer Mode
ABR	Available Bit Rate
CLR	Cell Loss Ratio
EOP	End of Packet
EPD	Early Packet Discard
FBA	Fair Buffer Allocation
FTP	File Transfer Protocol
GEO	Geosynchronous
GFR	Guaranteed Frame Rate
IP	Internet Protocol
MSB	Most Significant Bit
PPD	Partial Packet Discard
QoS	Quality of Service
RED	Random Early Discard
RFC	Request For Comment
SD	Selective Discard
SPEPD	Soft, Prioritized Early Packet Discard
TCP	Transmission Control Protocol
UBR	Unspecified Bit Rate
VPN	Virtual Private Network

TERMINOLOGY

Access Cards Input or Output Interface Cards of the Switch

CLAIMS

1. A packet discard system for controlling traffic congestion, within a buffered data switching network, said system comprising:

(a) a packet counter that begins to count the number of undiscarded packets after a packet discard event;

(b) an average buffer size counter;

(c) means for setting a packet count threshold responsive to the average buffer size counter; and

(d) means for performing a packet discard when the number of undiscarded packets reaches the packet count threshold.

2. A packet discard system as in claim 1, further comprising means for regularly calculating and updating average buffer size, using exponential averaging technique, thereby avoiding sudden changes in said average buffer size.

3. A packet discard system as in claim 1, wherein the packet count threshold is set progressively higher with increasing traffic congestion.

4. A packet discard system as in claim 1, wherein the packet counter is halted in a non-congested situation.

5. A packet discard system as in claim 1, further comprising means for applying a priority scheme for discarding packets.

6. A packet discard system as in claim 1, wherein the means for setting a packet count threshold uses a look-up table.

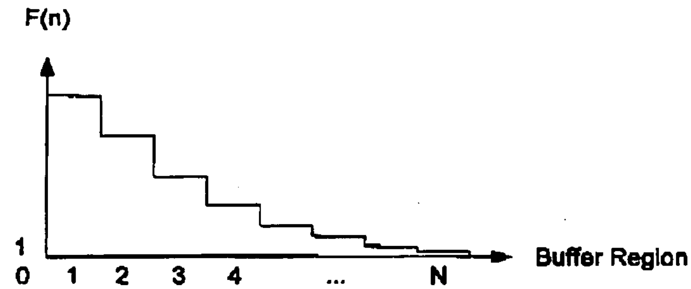


Figure 1

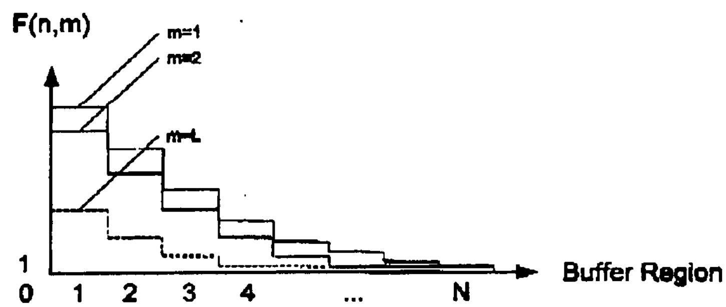
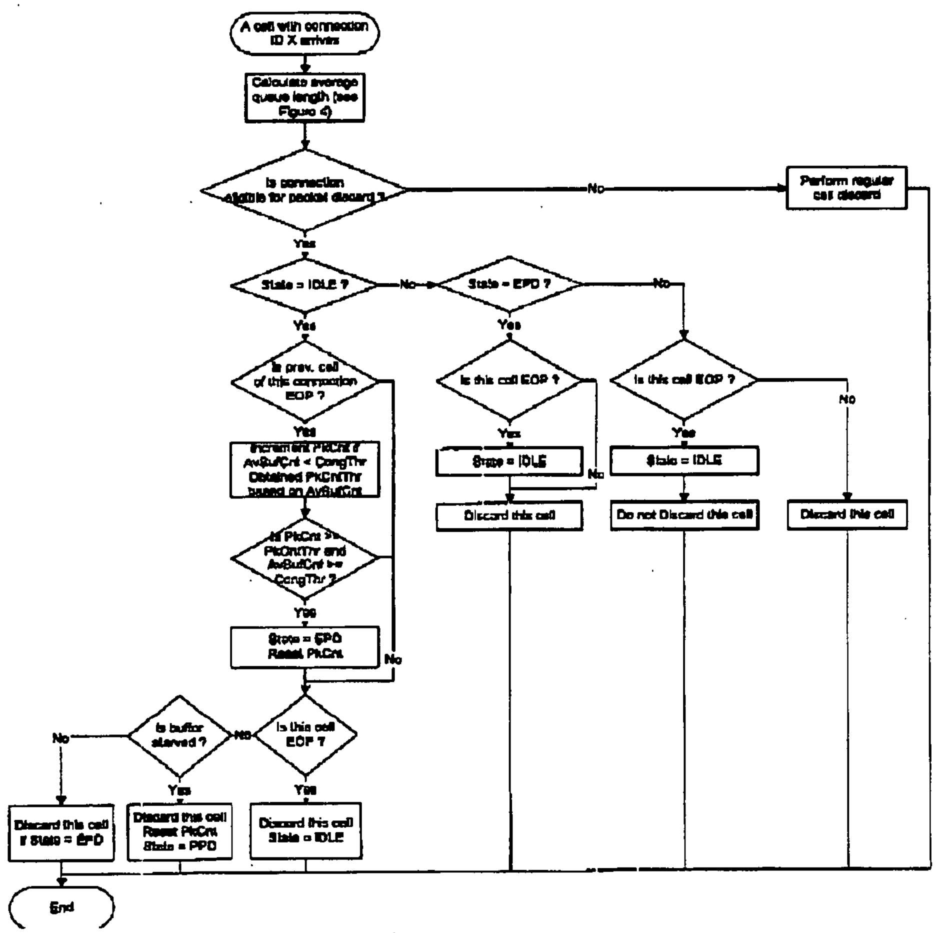


Figure 2

Borden Elliot Scott & Aylen


A cell with connection ID X arrives
Calculate average queue length (see Figure 4)
Is connection eligible for packet discard ?
No
Perform regular cell discard
Yes
State = IDLE ?
No
State = EPD ?
No
Yes
Is prev. cell of this connection EOP ?
Is this cell EOP ?
Is this cell EOP ?
No
Yes
Yes
State = IDLE
State = IDLE
No
Discard this cell
Do not Discard this cell
Discard this cell
Is PkCnt >= PkCntThr and AvBufCnt >= CongThr ?
Yes
State = EPD
Reset PkCnt
No
Is buffer starved ?
No
Is this cell EOP ?
No
Yes
Yes
Discard this cell if State = EPD
Discard this cell
Reset PkCnt
State = PPD
Discard this cell
State = IDLE
End
Figure 3

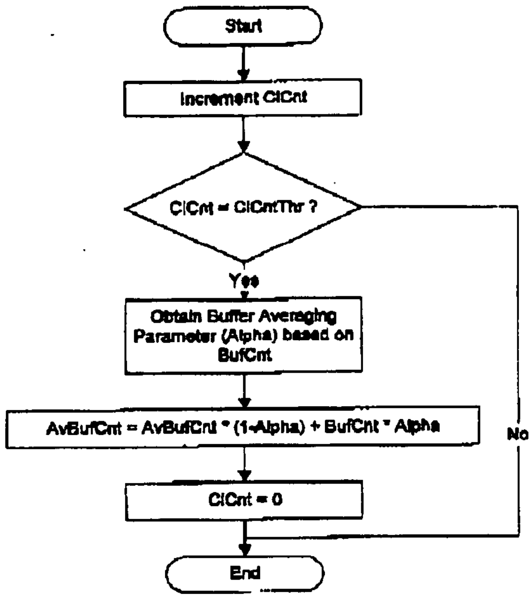


Figure 4

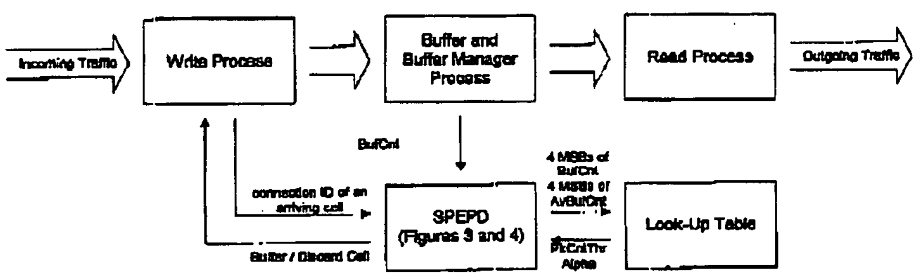


Figure 5

Borden Elliot Scott & Aylen

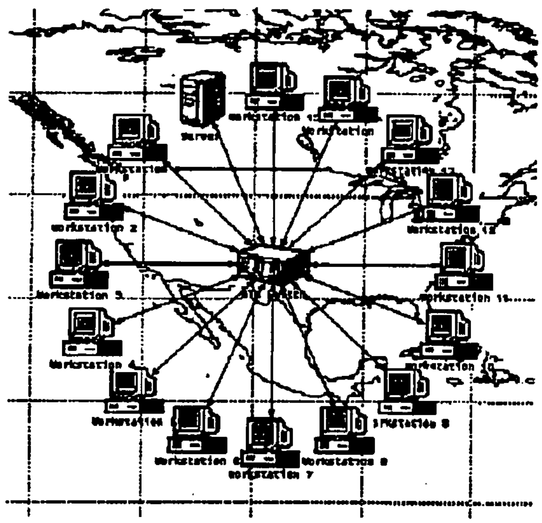


Figure 6

AvgBufCnt or BufCnt	PkCntThr	Alpha
0 % - 6.25 %	0	0.001953125
6.25 % - 12.5 %	0	0.001953125
12.5 % - 18.75 %	0	0.001953125
18.75 % - 25 %	0	0.001953125
25 % - 31.25 %	80	0.001953125
31.25 % - 37.5 %	70	0.001953125
37.5 % - 43.75 %	60	0.001953125
43.75 % - 50 %	50	0.001953125
50 % - 56.25 %	40	0.00390625
56.25 % - 62.5 %	30	0.00390625
62.5 % - 68.75 %	20	0.00390625
68.75 % - 75 %	10	0.00390625
75 % - 81.25 %	1	0.0078125
81.25 % - 87.5 %	1	0.0078125
87.5 % - 93.75 %	1	0.0078125
93.75 % - 100 %	1	0.0078125
CongThr :	25%	
ClCntThr :	210 cells	
Switch Buffer Size (per port) :	8192 cells	

Figure 7

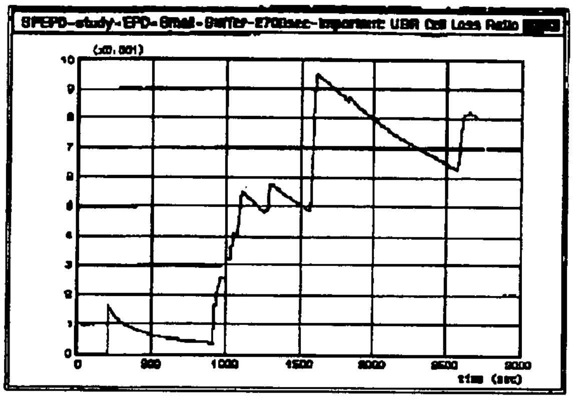


Figure 8

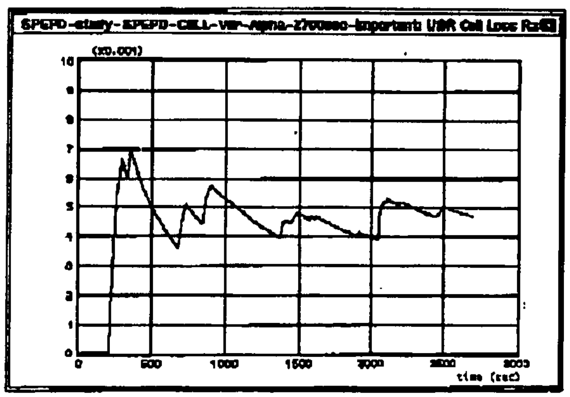


Figure 9

Borden Elliot Scott & Aylen

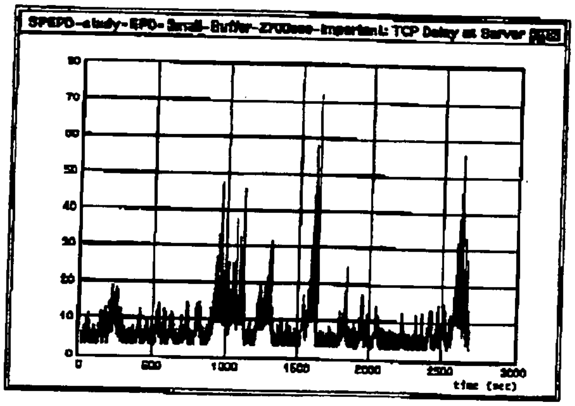


Figure 10

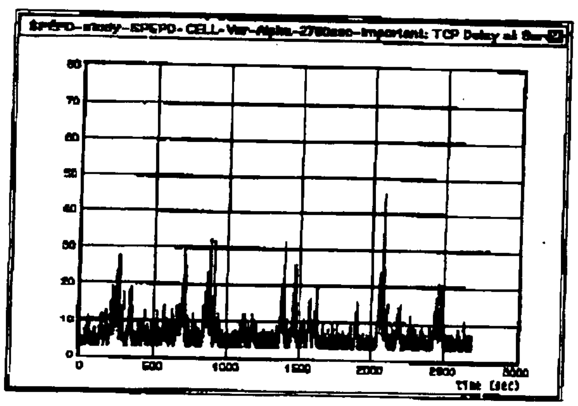


Figure 11

Borden Elliot Scott & Aylen